下载APP

30 | 四线复盘法：怎么避免成为背锅侠？

2021-02-08 李运华

大厂晋升指南 进入课程 >

讲述：安晓辉
时长 11:45 大小 10.77M

你好，我是华仔。

在事后总结阶段，正常情况下我们主要是做**收获总结**和**成果汇报**即可，但是如果发生了明显的问题，就需要做**问题复盘**。

复盘是一个围棋术语，它指的是对局结束后回顾记录，检查招法的优劣和得失关键，并且根据分析提出更好的招法，提升以后的对局能力。后来，这个思路被引入到了管理工作中。

问题复盘

技术人员主要参与的是线上问题复盘，比如业务或者系统出现了线上问题，在问题解决之后往往就会组织复盘。

不管团队技术多么厉害，也不管公司多么有钱，都不能完全避免业务或者系统出现问题的可能，比如 2015 年 5 月 27 日支付宝发生了大规模宕机的事故，2018 年 10 月 22 日 GitHub 发生了宕机 24 小时的事故等。

虽然无论做什么都不可能完全杜绝问题的发生，但这并不意味着我们只能坐以待毙。我们需要尽量降低问题发生的概率，减少问题导致的损失，因为就算事故不可避免，1 年发生 3 次和 10 年发生 1 次，影响和意义也是完全不同的。

问题复盘的意义就在于找到问题的原因然后加以改进，避免同样的问题反复出现，降低问题的发生的概率和影响。

四线复盘法

但是，要做好问题复盘可不是一件容易的事。复盘会议上的各种明争暗斗，可能会让刚参加工作的“萌新”惊掉下巴，甚至让一些老员工也感到头疼。尤其是一些管理比较严的公司还会通过复盘来明确责任分配和处罚措施，复盘会议的激烈程度往往不亚于电视剧中的宫斗场景。

所以，怎么组织一场复盘，怎么分配责任和避免背锅，已经成了职场人的一项生存必备技能。

问题复盘的内容涵盖**事实、分析、定责和改进**4 个部分，一次成功的问题复盘需要达成以下 4 个目标：

1. **讲清楚事实**：事实是复盘的基础，如果连事实都没有讲清楚就开始分析、定责和改进，无异于搭建空中楼阁，做得再漂亮也是没有意义的。
2. **全面且深入地分析**：首先需要保证没有遗漏问题，其次需要深入分析问题根因，否则以后问题还是会以其他方式反复出现。
3. **得出让各方心服口服的定责结论**：需要有明确的定责标准，避免拍脑袋定责，或者按照级别和关系来定责。

4. **制定可以落地的改进措施**：避免提出一些虚头巴脑的措施，看起来高大上，实际上却不知道怎么落地，后续也无法跟踪。

这一讲分享的**四线复盘法**，就是通过时间线、问题链、责任链和改进线这 4 条不同的线索来展开复盘，从而实现事实、分析、定责和改进这 4 个部分的目标。

如果你是复盘负责人，四线复盘法可以让你不偏不倚、公平公正地组织复盘；如果你是复盘参与人，它可以让你避免背不必要的黑锅。当然，如果出现问题确实是你的责任，它也不会教你怎么逃避责任，而是会告诉你怎么思考和改进。

接下来，我会针对每条线索逐一讲解说明。

第一条线：时间线

为了讲清楚事实，我们要明确**时间线**，也就是**问题发生的经过**，包括问题发现、问题处理过程中采取的各种关键措施、问题恢复的时间和问题影响的结果等。

其中，时间信息非常关键，因为它能够反映出问题发现速度、各项措施执行时间和团队响应效率等指标。比如，运维重启 30 台机器花了 1 小时，通常情况下这种处理效率肯定是有问题的。

第二条线：问题链

为了全面且深入的分析，我们要明确**问题链**，也就是**问题的传导路径**。

通常情况下，一个问题往往不是单一原因导致的，而是多个原因“碰巧”组合在一起所导致的，所以分析整个问题的传导路径，才能全面地了解产生问题的过程。

同时，针对单个问题的分析也不能浅尝辄止，而应该采用第 26 讲的“5W 根因分析法”深入分析，找到根本原因，这样才能为后续制定改进措施提供有效的指导。

问题链的路径逻辑有两类：业务流程和项目流程。

业务流程是指，端到端的业务处理的过程，分析的对象是各个关联的系统。

项目流程是指，端到端的项目开发的过程，分析的对象是项目各个阶段相关的人员，比如开发、测试、产品和运维等。

我们一般先采用业务流程的逻辑将问题定位到单个系统，然后再针对单个系统采用项目流程的方式将问题定位到具体的人或者流程中的某个步骤。

第三条线：责任链

为了得出让各方心服口服的定责结论，我们要明确**责任链**，也就是问题责任人之间的关系。

我们需要结合时间线中问题影响的结果、公司的故障定级标准和问题链的分析，最终确定哪些团队或个人应该承担责任，分别承担多大的责任，接受什么样的处罚。

之所以叫责任链，是因为一个问题的发生往往是整个流程上多个环节相关的人处理有问题，才会导致最终问题的发生。比如开发人员引入 bug，测试人员遗漏了测试，产品人员没有验收到，最终才会在上线后发现问题，这个环节中只要有一个环节把握住了，问题就不会发生。

定责是问题复盘中最棘手的部分，因为定责的结果会直接影响团队和个人的绩效，所以做到公平公正、让各方都心服口服是一项很大的挑战。

通常情况下，制定明确的定责标准有利于尽量减少争议，常见的标准包括以下 4 条：

1. **违反公司规章 / 制度 / 流程的承担主责**：比如公司规定必须要有灰度策略才能升级，某业务版本直接全量升级导致发生问题。
2. **出现重大纰漏的承担主责**：比如测试时漏测了某个常见的业务场景，导致上线后发生问题，测试承担主责，产品承担主责（因为上线前验收阶段没有发现问题），开发反而不一定承担责任（看具体的公司和团队要求）。
3. **问题源头承担主责**：比如 A 系统磁盘故障导致接口响应很慢且问题持续很长时间，从而进一步导致 B 系统对外响应也超时，这种情况下 A 系统应该承担主责，B 系统承担次责。
4. **问题放大者承担主责**：比如 A 系统磁盘故障导致接口响应很慢但只持续了几分钟，结果诱发了 B 系统的设计缺陷，导致 B 系统瘫痪超过 1 小时，这种情况下 B 系统应该承担

主责。

第四条线：改进线

为了制定可以落地的改进措施，我们要明确改进线，也就是问题的改进计划，包括具体措施、改进责任人和时间节点等。

（注：在这一讲中，**问题责任人**是指为问题承担责任的人，**改进责任人**是指负责落实改进措施的人，不一定是同一个。）

改进计划的思路来源于两个方面：时间线和问题链，通过时间线找到问题处理过程中不合理和可以优化的地方；通过问题链找到具体需要解决的问题。

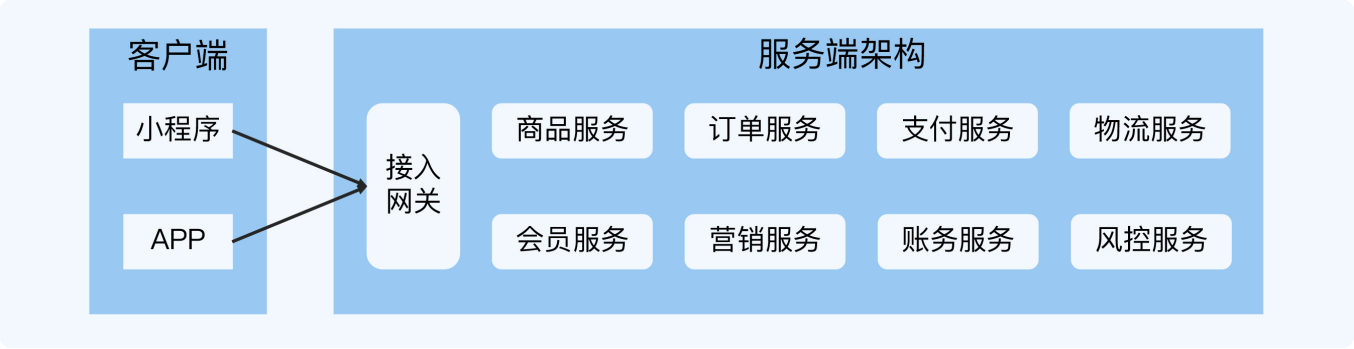
具体措施可以是流程上的调整（增加或删除流程步骤），技术上的手段（增加功能、优化系统）和团队方面的措施（学习、培训、奖惩机制）等。

无论采取什么措施，都要求能够落地执行。比如“提升团队质量意识”这种比较虚的措施，应该细化为“团队参加公司的质量规范学习和考试”“推行 Code Review”这种具体的措施。

接下来，我来带你拆解一个简单的线上问题复盘案例。

案例：线上商城

假设我们做了一个简单的线上商城，架构如下图所示：



某次线上故障导致用户下单后无法支付，我们按照四线复盘法来来复盘这个问题。

1. 时间线

首先，我们完整地回顾问题产生、处理和收尾的整个过程，梳理了时间线：

1. 07.24 10:23 风控服务进行了系统升级，支持新的风控策略。
2. 07.24 10:28 客服收到第1条用户投诉，投诉内容为：无法进行支付，客服引导用户尝试更换支付渠道，投诉内容解决。
3. 07.24 10:38 客服累计收到30条无法支付的投诉，根据客服管理条例，将问题升级，上报给业务值班长，业务值班长启动应急，召集订单服务、支付服务、会员服务、风控服务的负责人开始分析问题。
4. 07.24 10:50 支付服务的负责人抽查10个投诉用户的日志进行分析，发现是8个是因为风控拒绝支付导致的；风控负责人确认了风控返回支付拒绝，开始定位详细原因。
5. 07.24 11:30 风控负责人反馈暂时没有定位出详细原因，但今天系统进行了升级，很大概率和升级有关，建议回滚版本。
6. 07.24 11:40 风控服务相关的项目经理、产品经理、开发测试等人员决定回滚，开始回滚版本。
7. 07.24 12:10 风控服务回滚新版本完成，通知客服进行观察。
8. 07.24 12:30 客服反馈又收到了无法支付的投诉。
9. 07.24 12:40 支付服务和风控服务排查这次拒绝支付是正确的，建议客服继续观察。
10. 07.24 13:30 客服反馈没有新的无法支付的投诉。
11. 07.24 15:00 客服反馈只有1例无法支付的投诉，支付服务和风控服务排查是正常的风控拒绝。
12. 07.24 16:00 业务值班长、客服、各系统技术负责人讨论，确定本次应急结束。
13. 07.24 18:00 支付系统分析出受影响的用户数为2000人，实际放弃支付的人数为200人，涉及总金额为10000元；重试不同渠道后成功的有1800人。
14. 07.25 14:00 风控服务定位出问题根因后修复问题，重新升级版本。
15. 07.25 18:00 观察4小时，确认没有问题，问题解决。

2. 问题链

我们先按照业务流程来分析问题链，由于系统架构和这次问题都比较简单，所以问题链只涉及风控服务和支付服务：

1. 风控服务进行了升级，支持新的策略。新的策略通过配置指定不同银行渠道采取不同策略，升级的时候配置错误，将某个银行的配置值数字0误输入了字母O（因为键盘上两个键挨在一起）。因为数字0和字母O太像了，一开始排查的时候没有查出来。
2. 支付服务请求风控决策的时候，风控返回支付拒绝，导致支付失败。

针对风控服务的问题，我们再按照项目流程来分析问题链：

1. 运维人员升级风控服务。
2. 测试人员修改几十个配置，不小心将其中1个值的0输入成O了。
3. 产品验证的时候只挑选了几个大的银行渠道，正好没有验证到出问题的银行渠道。

3. 责任链

根据时间线中的影响结果，这次问题导致的损失是 10000 元；根据公司故障定级标准，属于轻微级别，惩罚措施是贡献活动经费；结合问题链和定责标准，我们得到了最终的责任链：

1. 风控服务承担主责，技术负责人贡献团队经费1000元；支付服务不承担责任。
2. 风控服务的测试人员承担主责，贡献团队建设经费500元；风控服务的产品人员承担次责，贡献团队建设经费200元。

4. 改进线

我们分析了时间线中的步骤，针对两个可以改进的地方制定了改进措施：

待改进点	改进措施	责任人	时间点	备注
风控服务定位问题从10:50开始到11:30结束，花费了40分钟，定位不出来才反馈可能和升级有关	应急流程明确出问题后第一时间明确是否有升级、修改配置、更换机器等操作，而不是先定位问题根因	业务值班长	2020.07.31	流程制度上的改进
风控服务的版本回滚从11:40开始到12:10结束，花费了30分钟，耗时有点长	优化版本回滚速度，将回滚耗时从30分钟降到5分钟内	风控服务技术负责人	2020.10.31	技术上的改进

然后，我们又分析了问题链中的问题，针对另外两个可以改进的地方制定了改进措施：

待改进点	改进措施	负责人	时间点	备注
测试人员修改几十个配置，不小心将其中1个的值0输入成O了。	系统增加配置有效值检查，包括值的类型、范围等，避免以来人工检查。	风控服务技术负责人	2020.09.31	通过技术手段来避免人的失误
产品验证的时候只挑选了几个大的银行渠道，正好没有验证到出问题的银行渠道。	所有渠道都需要验证，不能只抽查，如果人手不够，可以让测试人员一起验证	风控服务项目经理	2020.08.31	流程制度上的改进

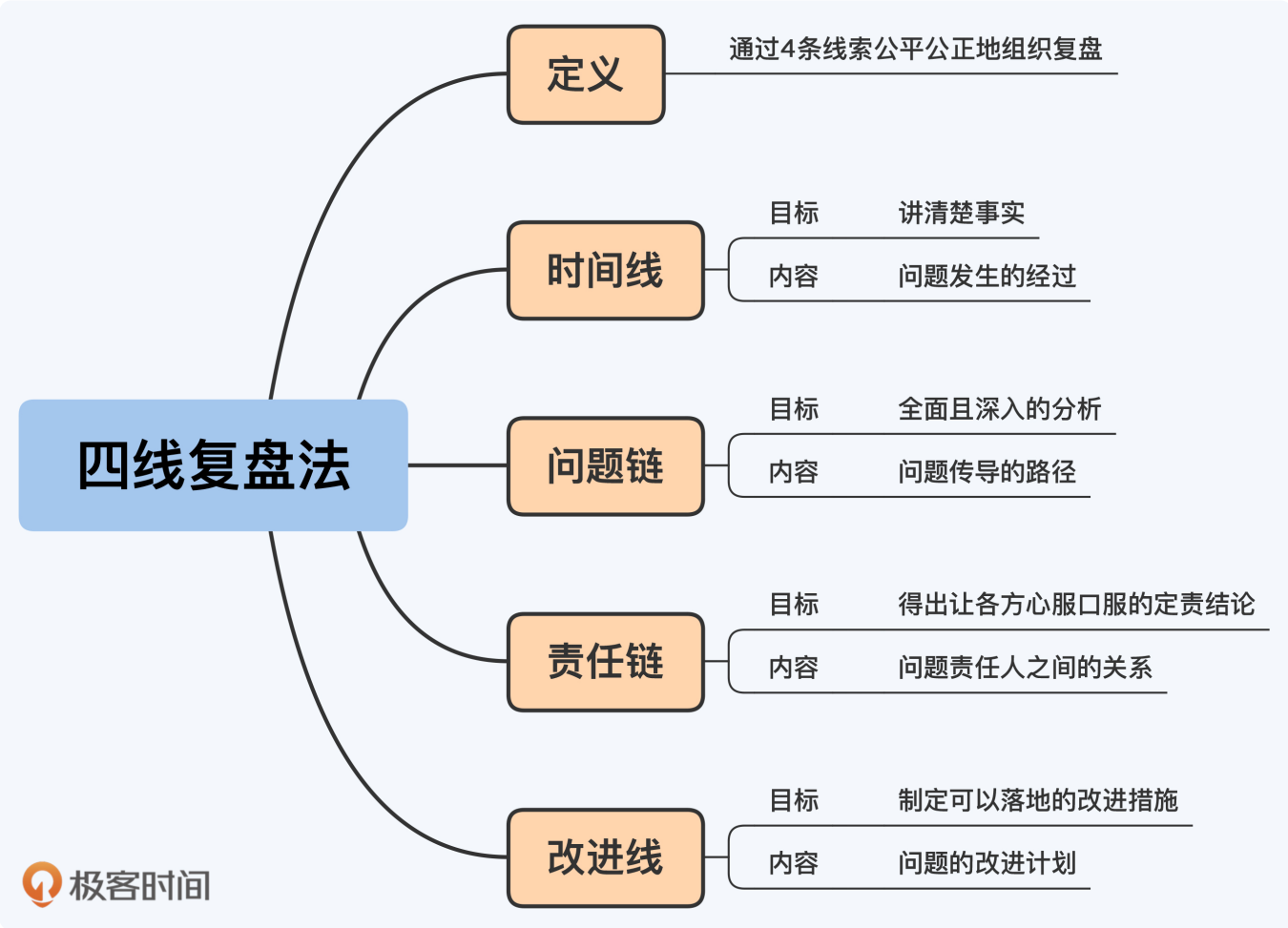
以上就是用四线复盘法对这次问题做复盘的整个过程。

小结

现在，我们回顾一下重点内容。

1. 一次成功的问题复盘需要达成 4 个目标：讲清楚事实，全面且深入地分析，得出让各方心服口服的定责结论，以及制定可以落地的改进措施。
2. 四线复盘法是通过时间线、问题链、责任链和改进线这 4 条不同的线索来展开复盘。它可以让你不偏不倚、公平公正地组织复盘，也可以让你避免背不必要的黑锅。

3. 时间线就是问题发生的经过，问题链就是问题的传导路径，责任链就是问题责任人之间的关系，改进线就是问题的改进计划。



思考题

这就是今天的全部内容，留一道课后思考题给你吧。你或者你的团队承担过线上问题的责任吗？如果有，主要原因是什么？你觉得处理结果是否公平，复盘过程有没有需要改进的地方？

欢迎你把答案写到留言区，和我一起讨论。相信经过深度思考的回答，也会让你对知识的理解更加深刻。

极客时间

宫斗撕逼不要怕，四线复盘不背锅。

李运华《大厂晋升指南》

提建议

更多课程推荐

乔新亮的CTO成长复盘

千万年薪 CTO 是如何炼成的

乔新亮
彩食鲜副总裁兼 CTO
前苏宁科技集团副总裁
TGO 鲲鹏会荣誉导师

涨价倒计时 现仅半价 ¥69，2月10日涨价至 ¥129

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 29 | 金字塔汇报法：怎么汇报才能让领导认可你的成果？

下一篇 31 | 导学：为什么业务和管理是晋升高级别的基石？

精选留言 (4)

写留言

菜心1986
2021-02-14

复盘什么频率做一次？复盘的归因和追责部分要到什么程度？
这里的追责结果是否代表个人本身有问题？或者本人能力不行？
多大的问题要追责，比如一周没事是否也要找个问题来汇报？
追责是否都是底层人人员，领导或者老板会有问题可以复盘么？

展开

作者回复: 1. 这里讲的是线上问题复盘，事件触发模式，不需要定期举行。
2. 多大问题要追责，实行什么样的奖惩措施，是公司或者团队要预先制定规则的。
3. 看问题大小来复盘和追责，支付宝5.27全网宕机的事故，总裁都要担责。

1

菜心1986
2021-02-14

这里的技术负责人都是底层研发吧。管理层，比如经理没有承担责任么？
另外，团队合作时，比如测试遗漏怎么定？比如需求文档没有明确要求，测试是不是不需要关心了，出了问题也是无责任？
或者这个因公司而异，都可以？

展开

作者回复: 1. 团队leader也要承担连带责任
2. 大部分公司以结果导向，即使需求文档没写，如果是正常需求需要考虑的，在设计和测试的时候都要覆盖，需求文档不可能事无巨细全部覆盖到。

1 1

王同学
2021-02-10

问题源头承担主责，问题放大者承担主责分不清楚。比如上游的数据格式出错，导致下游需要用数据的服务都有问题，那么是上游源头主责，还是下游放大问题者主责。上游定主责没问题，因为产生了错误的数据。下游定主责也可以，因为下游没有兼容缺陷数据。这个时候怎么定责

展开

作者回复: 你要看下游有没有放大问题，注意是放大，而不是说出错，因为上游数据格式出错，正常来说下游肯定出错，只要没放大就没问题。

qinsi

2021-02-09

“问题放大者承担主责”不是很理解。如果A是问题源头，B放大了问题，这时候是源头主责还是放大者主责呢?

展开

作者回复: B的主责肯定是跑不掉，至于A是不是主责，要看A造成的问题大小。举个例子，如果A的问题只持续了1分钟就解决了，那不需要承担主责，如果A本身也导致了30分钟的问题，那么也是要承担主责的。